SISTEMI OPERATIVI

ESERCIZIO del 8 APRILE 2005

In un **autolavaggio** vengono lavati due tipi di **veicoli**: **auto** e **camper**. L'autolavaggio può lavare più auto contemporaneamente. I camper possono essere lavati solo se non ci sono né auto né un altro camper in lavaggio, e hanno priorità sulle auto. Quando arriva un'auto, deve dare la precedenza agli eventuali camper in attesa.
Si implementi una soluzione usando il costrutto monitor per modellare l'**autolavaggio** e i processi per modellare i **veicoli** e si descriva la sincronizzazione tra i processi. Nella soluzione si massimizzi l'utilizzo delle risorse. Si discuta se la soluzione proposta può presentare starvation e in caso positivo per quali processi, e si propongano modifiche e/o aggiunte per evitare starvation.

```
program Autolavaggio

type      tipo = (auto, camper);  { tipo di veicolo }

type veicolo = process (t: tipo)
begin
      repeat
            a.entra (t);
            <esegui il lavaggio >
            a.esci (t);
      until false
end

type autola = monitor

{ variabili del monitor }
var  lav : array[tipo] of integer;
      { numero dei veicoli in lavaggio }
      coda : array[tipo] of condition;
      { code su cui sospendere i veicoli }

procedure entry entra (t: tipo)
begin
      { se è un'auto e c'è un camper in lavaggio o in coda }
      if t = auto and
            (lav[camper] > 0 or coda[camper].queue) then
            { sospensione }
            coda[t].wait;
      { se è un camper e c'è un veicolo in lavaggio }
      if t = camper and
            (lav[camper] > 0 or lav[auto] > 0) then
            { sospensione }
            coda[t].wait;
      { occupo la risorsa }
      lav[t]++;
end
```

```
procedure entry esci (t: tipo)
begin
      { rilascio le risorse }
      lav [t] --;
      { se è un camper }
      if t = camper
      begin
            { dà la precedenza ad una camper }
            if coda[camper].queue then
                  coda[camper].signal;
            else
                  while coda[auto].queue do
                        coda[auto].signal;
      end
      else { è un'auto }
      begin
            { se non ci sono più auto in lavaggio }
            if lav [auto] = 0 then
                  { risveglia un camper (se c'è) }
                  coda[camper].signal;
      end
end

begin { inizializzazione delle variabili }
      lav[auto] := 0;
      lav[camper] := 0;
end

var  a: autola; { il nostro monitor }
      a1, a2, ... : veicolo (auto);
      c1, c2, ... : veicolo (camper);

begin end.
```

Starvation

La soluzione proposta presenta starvation perché i camper possono ritardare il lavaggio delle auto in modo indefinito.

Si può ridurre il ritardo imponendo un contatore per i camper, alternando la priorità ogni tot di camper lavati.